Python Arrays

- items stored at contiguous memory locations.
- items must of the same type.
- Basic operations are Traverse, Insertion, Deletion, Search and Update
- variable_name = array(typecode,[elements])
- a = arr.array('i', [1, 2, 3]) # i=int, u=char, f=float
- Need to import Array Module - import array as arr

Python Array Operations

```
import array as arr
numbers = arr.array('i',[10,10,20,30])
numbers.append(50)
print(numbers.count(10))
numbers.extend([70,80,90])
numbers.insert(2,25)
numbers.pop(3)
numbers.remove(10)
print(numbers.index(10))
numbers.reverse()
print(len(numbers))
for number in numbers:
    print(number,end=" ")
lst = numbers.tolist()
print(lst)
```

Functions

```
We use functions to break up our code into small chunks.
These chunks are easier to read, understand and maintain.
------------------------
def sayHello():
    print("Hello")


sayHello()
```

Function - Passing Arguments

```
def add(a,b):  # Positional/Unnamed Arguments, must be in
order
    print(a+b)
add(2,3)
---------------------
def add(a,b): # keyword/Named Arguments
    return a+b
r=add(b=4,a=5)
print(r)
```

Keyword Args follow Positional Args

```
def add(c,a=0,b=0):
    return a+b+c
r=add(6,a=5,b=9)
print(r)
```

……………………………………….

```
def add(a=0,b=0,c):  #Error
    return a+b+c
r=add(6,a=5,b=9)
print(r)
```

Function - Default Arguments

```
def add(a=0,b=0):  # Positional/Unnamed Arguments, must be in
order
    print(a+b)
add(2,3)
add(2)
---------------------
def add(a=0,b=0): # keyword/Named Arguments
    return a+b
r=add(b=4,a=5)
r=add(b=4)
print(r)
```